

w.c.s. - Development #72802

faire le traçage des actions de workflow dans une table dédiée

27 décembre 2022 11:56 - Frédéric Péters

Statut:	Fermé	Début:	27 décembre 2022
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Aujourd'hui on ajoute des ActionsTracingEvolutionPart dans les demandes, ça les allourdit pour quelque chose utile uniquement pour la page /inspect.			
On pourrait stocker ça dans une table dédiée.			

Révisions associées

Révision 899fbadb - 02 janvier 2023 17:18 - Frédéric Péters

general: use a dedicated table to record workflow actions/events (#72802)

Historique

#2 - 30 décembre 2022 18:17 - Frédéric Péters

- Fichier 0001-general-use-a-dedicated-table-to-record-workflow-act.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

On stockait ActionsTracingEvolutionPart qui contenait l'événement (par exemple "création par import JSON") et les actions déclenchées. Je n'ai pas repris cette structure, plutôt un objet WorkflowTrace qui peut être soit un événement ("création par import JSON") soit une action (une seule).

L'enregistrement des événements est laissé à la responsabilité de l'endroit qui déclenche (plutôt que passer un argument "event" jusque perform_items, argument qui était à la base juste une chaîne puis qui dans certaines situations est devenu un tuple). Ça simplifie et rend les choses beaucoup plus claires, je trouve, mais ça faut deux appels, par exemple :

```
-         carddata.perform_workflow(event='json-import-created')
+         carddata.record_workflow_event('json-import-created')
+         carddata.perform_workflow()
```

Parfois deux appels prennent moins de place :

```
-         url = (
-             perform_items(
-                 action.items, formdata, event=('global-interactive-action', action.id), global_action=True
-             )
-             or url
-         )
+         formdata.record_workflow_event('global-interactive-action', action_id=action.id)
+         url = perform_items(action.items, formdata, global_action=True) or url
```

Pour l'enregistrement des actions, ça se fait, une par une, dans perform_items(). J'ai hésité à placer ça dans le perform() des actions mais ces méthodes ne font pas appel à super() et ça pouvait compliquer des tests unitaires d'action d'avoir ça exécuté systématiquement. Comme en plus il y a un seul endroit où perform() est appelé, ça allait très bien de juste ajouter la ligne de code là.

À propos des tests, quand même, il y en a une série qui appelaient perform_workflow (qui appelle perform_items) sur un formdata pas encore stocké en base. Comme on a besoin de l'id du formdata pour enregistrer la trace ça ne marche plus, il y a donc toute une série de formdata.store() ajoutés dans les tests. Cela concerne uniquement les tests, dans l'usage réel de w.c.s. il y avait déjà sauvegarde.

Il y a du code côté SQL (do_table/store/_row2ob) qui semble pouvoir être factorisé plus haut mais il y a parfois des petites différences, j'ai préféré rester à dupliquer un peu plutôt que mêler ce refactoring.

La migration demande de passer sur toutes les données, c'est quelque chose qu'on fait normalement de manière asynchrone (via set_reindex) mais comme on dépend de l'ordre d'enregistrement des lignes de trace dans la db c'est fait directement lors de la migration. Ça va sans doute prendre un

certain temps sur les grosses instances.

En faisant des tests je me suis rendu compte qu'on n'enregistrait pas de manière explicite de changement de statut lors du clic sur un bouton (d'une action "saut manuel"); on voyait le changement de statut simplement parce que des actions dans le statut qui suivait étaient exécutées. J'ai profité de ce moment pour ajouter la trace explicite (une ligne dans wcs/wf/choice.py).

#3 - 02 janvier 2023 16:00 - Frédéric Péters

- Fichier 0001-general-use-a-dedicated-table-to-record-workflow-act.patch ajouté

Version revue suite aux commentaires sur jabber :

- tri sur la colonne timestamp plutôt que sur l'id, ce qui permet d'exécuter la migration de manière asynchrone,
- modification aux noms des paramètres de détail pour être plus explicites (particulièrement : action_id devient soit global_action_id quand il s'agit de l'id d'une action globale, soit action_item_id quand il s'agit de l'id d'un "item" d'action).
- dans la foulée de ce changement, ajout de cette information à quelques endroits qui ne l'avaient pas.
- et également modification dans la db pour que les colonnes soient nommées action_item_id et action_item_key (plutôt que action_id et action_key)

#4 - 02 janvier 2023 16:56 - Lauréline Guérin

- Statut changé de Solution proposée à Solution validée

#5 - 02 janvier 2023 17:19 - Frédéric Péters

- Statut changé de Solution validée à Résolu (à déployer)

```
commit 899fbadb0909f9021f51090017b7048d1503d36c
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Fri Dec 30 10:04:45 2022 +0100
```

```
general: use a dedicated table to record workflow actions/events (#72802)
```

#6 - 02 janvier 2023 19:14 - Transition automatique

- Statut changé de Résolu (à déployer) à Solution déployée

#7 - 05 mars 2023 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-general-use-a-dedicated-table-to-record-workflow-act.patch	80,9 ko	30 décembre 2022	Frédéric Péters
0001-general-use-a-dedicated-table-to-record-workflow-act.patch	82,2 ko	02 janvier 2023	Frédéric Péters